

**WHAT IS CLAIMED IS:**

1. A method for optimizing instructions in a program, comprising:  
identifying first and second instructions in which (A) a first memory location and a first register contain the same value preceding the first instruction, (B) the first instruction stores a value into a second memory location, and (C) the second instruction subsequent to the first instruction loads a value from the first memory location into a second register;

between the first instruction and second instruction, inserting a third instruction which copies the value in the first register into the second register;

determining if the first memory location and the second memory location are the same memory location; and

conditionally executing the third instruction depending on a result of the determination.

2. A method according to claim 1, wherein the first memory location and the first register are determined to contain the same value because a preceding instruction either loaded the register from the memory location or stored the register to the memory location.

3. A method according to claim 1, wherein the second instruction is executed if the first and second memory locations are the same.

4. A method according to claim 1, wherein the second instruction is nullified if the first and second memory locations are different.

5. A method according to claim 1, wherein the insertion of the third instruction is done during compile-time.

6. A method according to claim 1, wherein the insertion of the third instruction is done during run-time.

7. A method according to claim 1, wherein the determination is made during run-time.

8. A method according to claim 1, wherein the inserting further comprises inserting a fourth instruction between the third instruction and the second instruction which compares the first memory location and the second memory location.

9. A method according to claim 8, wherein the fourth instruction nullifies the second instruction if the first memory location and the second memory location are different.

10. A method for optimizing instructions in a program, comprising:  
identifying first and second instructions in which (A) a first memory location and a first register contain the same value preceding the first instruction, (B) the first instruction stores a value into a second memory location, and (C) the second instruction subsequent to the first instruction loads a value from the first memory location into a second register;

replacing the second instruction such that, a run-time determination is made as to whether a load into the second register would be redundant, in which case, a register copy, rather than a load from memory, is executed.

11. A method according to claim 10, wherein the run-time determination is a determination as to whether the first and second memory locations are the same.

12. A method according to claim 10, wherein the second instruction is replaced by a register copy, a test of the identity of the first and second memory locations, and a load from memory whose execution is conditioned on the test.

13. A computer readable medium on which is stored software for optimizing instructions in a program, including software to direct a computer to:

identify first and second instructions in which (A) a first memory location and a first register contain the same value preceding the first instruction, (B) the first instruction stores a value into a second memory location, and (C) the second instruction subsequent to the first instruction loads a value from the first memory location into a second register;

between the first instruction and second instruction, insert a third instruction which copies the value in the first register into the second register;

determine if the first memory location and the second memory location are the same memory location; and

conditionally execute the third instruction depending on a result of the determination.

14. A computer readable medium according to claim 13, wherein the first memory location and the first register are determined to contain the same value because a preceding instruction either loaded the register from the memory location or stored the register to the memory location.

15. A computer readable medium according to claim 13, wherein the second instruction is executed if the first and second memory locations are the same.

16. A computer readable medium according to claim 13, wherein the second instruction is nullified if the first and second memory locations are different.

17. A computer readable medium according to claim 13, wherein the insertion of the third instruction is done during compile-time.

18. A computer readable medium according to claim 13, wherein the insertion of the third instruction is done during run-time.

19. A computer readable medium according to claim 13, wherein the determination is made during run-time.

20. A computer readable medium according to claim 13, wherein the inserting further comprises inserting a fourth instruction between the third instruction and the second instruction which compares the first memory location and the second memory location.

21. A computer readable medium according to claim 20, wherein the fourth instruction nullifies the second instruction if the first memory location and the second memory location are different.